Acta Crystallographica Section A
Foundations of Crystallography
ISSN 0108-7673

Fast differentiation algorithm and efficient calculation of the exact matrix of second derivatives

Alexandre G. Urzhumtsev and Vladimir Y. Lunin

Acta Crystallographica Section A
Foundations of Crystallography
ISSN 0108-7673

Received 8 November 2000
Accepted 21 March 2001

Fast differentiation algorithm and efficient calculation of the exact matrix of second derivatives

Alexandre G. Urzhumtsev[a]* and Vladimir Y. Lunin[a,b]

[a]LCM3B, UMR 7036 CNRS, Faculté des Sciences, Université Henri Poincaré, Nancy I, 54506 Vandoeuvre-les-Nancy, France, and [b]Institute of Mathematical Problems of Biology, Russian Academy of Sciences, Pushchino, Moscow Region 142290, Russia. Correspondence e-mail: sacha@lcm3b.uhp-nancy.fr

Many crystallographic problems are reduced to the optimization of some functional. In most cases, this functional is expressed in terms of structure factors and depends on a large number of variables; an example is the refinement of atomic models. Calculation of the functional derivatives, necessary for different optimization methods, is a time-consuming procedure. Previously, a technique to calculate the exact gradient of any crystallographic functional for the time equal to that for a single-function-value calculation has been proposed [Lunin & Urzhumtsev (1985). *Acta Cryst.* A**41**, 327–333]. Currently, a similar scheme is proposed to calculate the exact matrix of the second derivatives of these functionals. The accuracy of this matrix is crucial for the calculation of the inverted matrix, which can be used in optimization methods of the second order.

1. Notation

In what follows, vectors are presented by their coordinates in columns and T stands for the transposition of vectors and matrices.

$\mathbf{x} = (x_1, \ldots, x_N)^T$: N-dimensional vector of parameters;
$\mathbf{y} = (y_1, \ldots, y_M)^T = [y_1(x_1, \ldots, x_N), \ldots, y_M(x_1, \ldots, x_N)]^T$: M-dimensional vector function of parameters $x_1, \ldots, x_N$;
$R(y_1, \ldots, y_M)$: function of variables $y_1, \ldots, y_M$;
$\nabla_{\mathbf{x}} R = (\partial R/\partial x_1, \ldots, \partial R/\partial x_N)^T$, $\nabla_{\mathbf{y}} R = (\partial R/\partial y_1, \ldots, \partial R/\partial y_M)^T$: gradients of the function R with respect to variables $\mathbf{x}$ and $\mathbf{y}$;

$$[\mathrm{d}\mathbf{y}/\mathrm{d}\mathbf{x}] = \begin{pmatrix} \partial y_1/\partial x_1 & \partial y_2/\partial x_1 & \ldots & \partial y_M/\partial x_1 \\ \partial y_1/\partial x_2 & \partial y_2/\partial x_2 & \ldots & \partial y_M/\partial x_2 \\ \vdots & \vdots & \vdots & \vdots \\ \partial y_1/\partial x_N & \partial y_2/\partial x_N & \ldots & \partial y_M/\partial x_N \end{pmatrix};$$

$$[H] = \Delta_{\mathbf{x}} R = \nabla^2_{\mathbf{x}} R$$
$$= \begin{pmatrix} \partial^2 R/\partial x_1 \partial x_1 & \partial^2 R/\partial x_2 \partial x_1 & \ldots & \partial^2 R/\partial x_N \partial x_1 \\ \partial^2 R/\partial x_1 \partial x_2 & \partial^2 R/\partial x_2 \partial x_2 & \ldots & \partial^2 R/\partial x_N \partial x_2 \\ \vdots & \vdots & \vdots & \vdots \\ \partial^2 R/\partial x_1 \partial x_N & \partial^2 R/\partial x_2 \partial x_N & \ldots & \partial^2 R/\partial x_N \partial x_N \end{pmatrix}.$$

2. Introduction

Further development of the software for the refinement of atomic molecular models includes several aspects like elaboration of convenient user-friendly interfaces, suggestions for new refinement criteria and use of more sophisticated optimization algorithms. Recently, several groups reported some advances in the calculation of an approximate matrix of the second derivatives (Hessian or normal matrix) for crystallographic criteria (Murshudov *et al.*, 1997; Templeton, 1999; Tronrud, 1999), which can eventually be used in optimization methods of the second order. These works uncover several questions. First, these algorithms have been developed for some particular criteria and it would be interesting to generalize them if possible. Then, the second-order minimization methods need an inverted Hessian matrix with an inversion that is very sensitive to approximations; as a consequence, algorithms for the calculation of the *exact* Hessian matrix must be developed in this case. Finally, crystallographic refinement uses combined minimization criteria and the algorithms developed should be completed by the algorithms to calculate such additional contributions from geometric and other terms.

Known methods for the optimization (in what follows only the minimization will be considered) of functions in multidimensional space do not allow their minima to be found directly from the starting point and are iterative: they need to verify the function value in several points of the search space before arriving at the result. These methods can be divided in groups according to the order of the derivatives of the function used during the search. The higher the order of the calculated derivatives, the more information is used, and the number of iterations necessary to find the solution (or to approach it equally well) is smaller. However, the computer time necessary to carry out every minimization cycle becomes larger and a choice of an optimal method depends on each case.

The methods that are based on the function values only (exhaustive search, Monte Carlo, simplex method, coordinate descent, simple iteration *etc.*) check the function value in a set of points chosen in some way, either random or initially prescribed.

A series of gradient minimization methods is based on the iterative recalculation of the descent direction as a linear combination of the antigradient with the previously defined directions and a parameter shift along them. For the conjugate-gradient methods, these directions are orthogonal or, in the general case, conjugated (orthogonal with a given matrix), giving the name for this group of methods.[1] In a space of a finite dimension N, such a property is a guarantee that the result will be found by N iterations.

Let $\mathbf{q}_n$ stand for the point in the parameter space found after the nth iteration, $\mathbf{r}_n$ stands for the corresponding gradient of the function R to be minimized, $\mathbf{s}_n$ stands for the corresponding direction of a one-dimensional minimization and $\mathbf{d}_n$ stands for the correction of the parameters at this iteration. The basic formulae for the conjugate-gradient method contain the Hessian matrix $[H]$ used to calculate the correction coefficient β_n and the optimal step α_n:

$$\begin{array}{ll}
\text{Initial point} & n=0;\ \mathbf{q}_0;\ \mathbf{s}_0=\mathbf{0};\ \beta_0=0\\
\text{Gradient in the current point} & \mathbf{r}_n=-\nabla R(\mathbf{q}_n)\\
\multicolumn{2}{l}{\text{Correction coefficient for the search direction}}\\
 & \beta_n=\mathbf{r}_n^T[H]\mathbf{s}_n/\mathbf{s}_n^T[H]\mathbf{s}_n\\
\text{Search direction} & \mathbf{s}_{n+1}=\mathbf{r}_n-\beta_n\mathbf{s}_n\\
\multicolumn{2}{l}{\text{Optimal step along the search direction}}\\
 & \alpha_{n+1}=\mathbf{s}_{n+1}^T\mathbf{r}_n/\mathbf{s}_{n+1}^T[H]\mathbf{s}_{n+1}\\
\text{Parameter shift} & \mathbf{d}_{n+1}=\alpha_{n+1}\mathbf{s}_{n+1}\\
\multicolumn{2}{l}{\text{New point in the search space (parameter values)}}\\
 & \mathbf{q}_{n+1}=\mathbf{q}_n+\mathbf{d}_{n+1}.
\end{array}\tag{1}$$

However, one does not need to calculate explicitly all N^2 components of the matrix $[H]$ but only its product by a vector, *i.e.* N values (see §§4 and 5 below).

Naturally, for non-quadratic functions, (1) gives only estimations for the optimal coefficients and the step. A number of alternative approximations are known and the results obtained with different formulae are exactly the same for quadratic functions but can be different in the general case. A comparison of some gradient methods for crystallographic refinement of atomic models has been performed by Tronrud (1992).

The knowledge of the second derivatives (Hessian or normal matrix) allows for a quadratic function of any number of variables to get the answer from any starting point without iterations. Methods based on the second derivatives like the Newton method use the approximation of a function by a quadratic one but here minimization again became iterative. In these methods, one does not need the matrix itself but the product of the inverse matrix by a vector (gradient of the function). Therefore, the computational price of every step of the Newton method seems to be much higher because of these operations.

Summarizing, it can be noted that the matrix of the second derivatives can be used for the minimization of a functional in three different aspects:

(i) in gradient methods in order to calculate the direction of the minimization;

(ii) in gradient methods in order to estimate the step of the displacement along a chosen direction in an N-dimensional space (one-dimensional search);

(iii) in minimization methods of the second order or higher;

while its explicit calculation is necessary only for the latter. There exist some other applications of the Hessian matrix, for example its use as a source of the accuracy of information. Neither these applications nor the problem of the matrix inversion are discussed here.

3. Latest developments

Owing to a very high dimensionality of the space of parameters for macromolecular atomic models, their refinement is usually performed by gradient methods, at least at the end of refinement. However, some refinement programs, developed originally for small molecules (for example, Hansen & Coppens, 1978; Sheldrick & Schneider, 1997) do use the methods of second order where the direct-matrix calculation needs of the order of MN^2 operations (derivative of the contribution of every reflection with respect to every couple of parameters). Here N is the number of atomic parameters and M is the number of structure factors. In order to gain CPU time, some authors use a sparse matrix (Sussman *et al.*, 1977; Hendrickson & Konnert, 1980; Dodson, 1981; Guillot *et al.*, 2001). A fast calculation of the full matrix of the second derivatives for the crystallographic criteria with the atomic parameters (coordinates, temperature factors, occupancies) as independent variables were discussed recently. One of these works (Templeton, 1999) is based on a statistical approach and the two others (Murshudov *et al.*, 1997; Tronrud, 1999) developed the FFT technique suggested for this goal by Agarwal (1978).

In the case of the standard crystallographic criterion, weighted least-squares fit of the magnitudes

$$R(\mathbf{q})=\tfrac{1}{2}\sum_{\mathbf{h}}w(\mathbf{h})[|F(\mathbf{q};\mathbf{h})|-F_o(\mathbf{h})]^2 \tag{2}$$

with atomic parameters $\mathbf{q}$ as independent variables, Tronrud (1999) proposed a practical algorithm that needs

$$T_{HT}=C_1N^2+C_2M\ln M \tag{3}$$

operations to calculate the principal part $S_1^{(jk)}$ of the matrix

[1] It can be noted that, in the original conjugate-gradient method (Lanczos, 1952; Hestenes & Stiefel, 1952), the conjugated vectors are the minimization directions and not the gradients, which are simply orthogonal.

$$\begin{aligned}\partial^2 R/\partial q_j \partial q_k &= \sum_{\mathbf{h}} w(\mathbf{h}) \partial |F(\mathbf{q};\mathbf{h})|/\partial q_j \partial |F(\mathbf{q};\mathbf{h})|/\partial q_k \\ &\quad + \sum_{\mathbf{h}} w(\mathbf{h})[|F(\mathbf{q};\mathbf{h})| - F_o(\mathbf{h})]\partial^2 |F(\mathbf{q};\mathbf{h})|/\partial q_j \partial q_k] \\ &= S_1^{(jk)} + S_2^{(jk)} \end{aligned} \tag{4}$$

traditionally neglecting the second term $S_2^{(jk)}$ with $\partial^2|F(\mathbf{q};\mathbf{h})|/\partial q_j \partial q_k$. Here again N is the number of atomic parameters, M is the number of structure factors, and C_1 and C_2 are universal constants that depend neither on N nor on M. Another important feature of Tronrud's (1999) algorithm is that it allows calculation of the matrix element by element and does not need to keep the whole matrix (N^2 elements) at the same time in computer memory. Similar formulae were derived by Murshudov *et al.* (1997) for the case of the maximum-likelihood (ML) criterion.

The following questions arise:

(i) which are the features allowing such fast calculation of the matrix in these cases?

(ii) is the suggested way of matrix calculation valid for other criteria depending on structure factors?

(iii) can better results be obtained in some particular cases? More questions arise from the fact that the matrix inversion is, in general, very sensitive to different approximations, especially when some eigenvalues are close to 0, which is the case (Cowtan & Ten Eyck, 1998; Ten Eyck, 1999). In consequence, the approximate matrices discussed above could be used for different estimations, however, in order to use correctly the second-order minimization methods, the *exact* matrices should be available (unless an algorithm is developed to calculate directly the inverted matrix without the calculation of the direct one). These extra questions are:

(iv) is it possible to calculate rapidly the *exact* matrix of the second derivatives with respect to atomic parameters for a least-squares criterion?, for the ML criterion?, in the general case?

(v) is this possible in the general case when the independent parameters are not the atomic coordinates but some generalized ones?

4. Fast differentiation algorithm

Traditionally, it is considered that gradient methods are much more time consuming than the methods without derivatives, and that generally every computation of a gradient needs about N times CPU more than a single value of the function, N being the number of variables. However, several groups (*e.g.* Baur & Strassen, 1983; Kim *et al.*, 1984) independently have shown the following result:

Fast differentiation algorithm (FDA). For any function of N variables, calculation of a single value of which takes the time T, an algorithm exists to calculate its *exact* gradient faster than $4T$, *independent of the number N* of variables.

The scheme proposed gives not only the time estimation but a practical technique of how a fast algorithm for the gradient, calculated without any approximation, can be produced from an algorithm for the function calculation. The basic idea is essentially that one should not try to derive direct relations linking the minimized functional and the initial parameters; on the contrary, it is necessary to calculate the derivatives using the chain rule, descending on the way back exactly by the same intermediate steps that were used in the direct method of the function calculation.

The constant '4' is a upper theoretical limit estimated for the worst case. In practice, this constant approaches 1. Since many expressions are exactly the same for the calculation of the function and of its gradient, the use of computer memory can save more CPU time and practical estimations with the program *FROG* (Urzhumtsev *et al.*, 1989) showed that the total time to calculate the function and its gradient is $1.5T$ and not $5T$ (T for the function calculation plus $4T$ for its gradient).

From the FDA, the following conclusion can easily be derived:

FDA: conclusion 1. For any function of N variables, calculation of a single value of which takes the time T, an algorithm exists to calculate its *exact* derivative along a given direction faster than $4T$, *independent of the number N* of variables and *without* a gradient calculation.

For gradient minimization procedures, this observation allows, as was realised in *FROG*, at any point the calculation of the function and its derivative along the search direction and therefore to approximate better (by a polynomial of the third degree) the function and eventually to avoid some local minima.

Since the product of the matrix of the second derivatives $[H]$ in a given direction $\mathbf{s}$ can be interpreted as a gradient of the scalar product of two vectors, the given direction $\mathbf{s}$ and the gradient ∇f:

$$[H]\mathbf{s} = \nabla(\nabla f \cdot \mathbf{s}), \tag{5}$$

the following conclusion relevant to the problem of this paper is evident:

FDA: conclusion 2. For any function of N variables, calculation of a single value of which takes the time T, an algorithm exists to calculate the *exact* product of the matrix of the second derivatives by a given direction faster than $20T$, *independent of the number N* of variables.

Again, in practice, when some common expressions can be saved in memory, the time to calculate all four objects, namely the function f itself, its gradient ∇f, the function derivative $\partial f/\partial \mathbf{s}$ along a given direction $\mathbf{s}$ and the product $[H]\mathbf{s}$ of the Hessian matrix $[H]$ in the same direction, can be estimated as $4T$, where T stands for the time of a single calculation of the function value.

5. The FDA and crystallographic refinement

5.1. Initial scheme by Sayre–Ten Eyck–Agarwal

As follows from §4, the crucial point in the fast optimization of a crystallographic functional is a fast calculation of structure factors. The original idea by Sayre (1951) of an intermediate use of the electron density instead of direct calculation of structure factors from the atomic parameters was developed and realised later by Ten Eyck (1977), and this drastically changed macromolecular crystallographic computing.

In this scheme, the first step of density generation from an atomic model does not depend on the number of structure factors (this transition from atomic parameters to the electron density 'does not know' how this density will be used later) and is linear with the number N of atoms. The second step of structure-factor calculation, on the contrary, 'does not know' how the density was calculated and the amount of calculations is practically linear with the number M of reflections. In total, about

$$T_f = C_1 N + C_2 M \ln M \qquad (6)$$

operations is necessary to calculate a set of structure factors and, as a consequence, a single value of the least-squares criterion (C_1 and C_2 are some constants), instead of CNM operations for a direct calculation. Different corrections, for example, for bulk solvent can easily be included in this way. It can be noted also that such a fast way of calculation is quite natural since X-rays are diffracted by electrons distributed in the unit cell and not by pointed atoms.

An application of the FDA to the least-squares criterion depending on atomic parameters gave immediately an improved version (Lunin, 1978, unpublished; Lifchitz, in Agarwal, 1981) of Agarwal's original fast algorithm for gradient calculation (Agarwal, 1978) and suggested a general scheme for the development of the gradient-based refinement programs (Lunin & Urzhumtsev, 1985; Urzhumtsev *et al.*, 1989). The FDA shows also that for complicated and time-consuming problems like crystallographic refinement one needs to propose only a fast way of function calculation without having the gradient problem in mind.

5.2. General scheme

This scheme can be easily extended to the general case when independent model parameters $\boldsymbol{\chi}$ are no longer atomic coordinates but some other diffraction parameters (*e.g.* model description by rigid groups) and the criterion is not necessarily the least-squares fit of structure-factor magnitudes. The parameters used for crystallographic calculations can be considered as parameters of structure description at different levels (Lunin & Urzhumtsev, 1985; Urzhumtsev *et al.*, 1989): $\boldsymbol{\chi}$ generalized parameters; $\mathbf{q}$ atomic model; $\boldsymbol{\rho}$ electron density; $\mathbf{F}$ structure factors.[2] An efficient function calculation should pass therefore through the steps

$$\boldsymbol{\chi} \rightarrow \mathbf{q} \rightarrow \boldsymbol{\rho} \rightarrow \mathbf{F} \rightarrow R(\mathbf{F}). \qquad (7)$$

This algorithm also allows the contribution of criteria of other types to be added, for example stereochemical ones or those depending on the electron-density distribution or on the generalized parameters (*e.g.* rigid-group orientation or position). As a consequence of this scheme and the FDA, the fast calculation of the gradient of the total function should pass again through these steps but in inverted order:

$$R(\mathbf{F}) \rightarrow \nabla_{\mathbf{F}} R \rightarrow \nabla_{\boldsymbol{\rho}} R \rightarrow \nabla_{\mathbf{q}} R \rightarrow \nabla_{\boldsymbol{\chi}} R. \qquad (8)$$

In other words, the gradient calculation demands the consecutive recalculation of the gradient with respect to the variables of such a chain and *not* its direct calculation with respect to the original variables. The contributions of other additional criteria are included automatically in the scheme (8) (Urzhumtsev *et al.*, 1989).

The FDA provides immediately the following results:

(i) for *any* additive criterion expressed in the terms of structure factors, its gradient with respect to atomic parameters can be calculated for the time $C_1 N + C_2 M \ln M$;

(ii) for *any* additive criterion expressed in terms of structure factors, the product of its matrix of the second derivatives by any given vector in the space of atomic parameters can be calculated for the time $C_1 N + C_2 M \ln M$;

(iii) if the model is described by generalized parameters and the recalculation of the atomic parameters from them needs G operations, then the gradient of *any* additive structure-factor criterion with respect to the generalized parameters and the product of its matrix of the second derivatives by any given direction in the space of the generalized parameters can be calculated by $C_3 G + C_1 N + C_2 M \ln M$ operations, where C_3 is a small number ($C_3 \sim 4$);

(here a criterion is called *additive* if it is represented by a sum of contributions from every parameter of this level). This latter result is important for conjugate-gradient methods, which need the calculation of the product of the matrix of the second derivatives by the direction of the previous descent; as has been shown above, this product can be calculated exactly and practically for the same time as a single value of the function.

5.3. Application to the calculation of the matrix of second derivatives

Another important consequence of the FDA for crystallographic refinement is the following. The nth column of the matrix $[H]$ of the second derivatives represents the product of this matrix by the direction $(0, 0, \ldots, 0, 1, 0, \ldots, 0)$, where 1 is in the nth position. This means that the whole *exact* matrix of the second derivatives composed from N columns (the case of any additive structure-factor functional and a model with independent atomic parameters) can be calculated by

$$T_{HE} = C_1 N^2 + C_2 N M \ln M \qquad (9)$$

operations (and K-times faster if the matrix is block diagonal composed of K blocks of roughly equal size). Comparison of this estimation with (3) obtained by Tronrud (1999) shows that the 'price' of the calculation of the *exact* matrix instead of its

[2] Eventually, more levels can be introduced. For example, crystallographic images calculated with a limited set of structure factors at a given resolution are not the same as the exact electron density $\boldsymbol{\rho}$ calculated at the third level and can be considered as a structure parameterization at the level $\mathbf{J}$, after the level $\mathbf{F}$. For example, a histogram comparison can be applied to such an object and not to an object of the level $\boldsymbol{\rho}$.

approximation is the replacement of C_2M by C_2MN. Usually M is of the order of N, at least for the macromolecules, and due to the presence of the member C_1N^2 this replacement does not mean the multiplication of the whole amount of operations T_{HT} by N but rather only its duplication even in such a non-optimized way for T_{HE}. (Note that this procedure does not need to have at any moment the whole matrix $[H]$ composed of $N \times N$ elements and can always operate only with vectors of the length N, columns of this matrix.) A faster and more general way to calculate the exact Hessian matrix is discussed below.

6. The chain rule and the matrix of second derivatives

6.1. General analysis

Let a function $R(y_1(x_1, \ldots, x_N), \ldots, y_M(x_1, \ldots, x_N))$ depend on M variables $y_1, \ldots, y_M$ with every y_m depending in turn on N variables $x_1, \ldots, x_N$. The chain rule

$$\partial R/\partial x_j = \sum_m [(\partial R/\partial y_m)(\partial y_m/\partial x_j)] \tag{10}$$

and its consequence

$$\begin{aligned}\partial^2 R/\partial x_j \partial x_k = &\sum_m \sum_n [(\partial^2 R/\partial y_m \partial y_n)(\partial y_n/\partial x_k)(\partial y_m/\partial x_j)] \\ &+ \sum_m [(\partial R/\partial y_m)(\partial^2 y_m/\partial x_j \partial x_k)]\end{aligned} \tag{11}$$

give

$$\boldsymbol{\Delta}_{\mathbf{x}} R = [\mathrm{d}\mathbf{y}/\mathrm{d}\mathbf{x}][\boldsymbol{\Delta}_{\mathbf{y}} R][\mathrm{d}\mathbf{y}/\mathrm{d}\mathbf{x}]^T + [\boldsymbol{\Delta}_{\mathbf{x}}\mathbf{y}]\nabla_{\mathbf{y}} R, \tag{12}$$

which can be used for the recalculation of the Hessian matrix with respect to variables $\mathbf{x}$ when necessary derivatives are known for the variables $\mathbf{y}$ of the superior level. Here $[\boldsymbol{\Delta}_{\mathbf{x}}\mathbf{y}]$ is a tensor composed of M matrices $[\boldsymbol{\Delta}_{\mathbf{x}} y_1]$, $[\boldsymbol{\Delta}_{\mathbf{x}} y_2]$, $\ldots$, $[\boldsymbol{\Delta}_{\mathbf{x}} y_M]$. The gradient $[\nabla_{\mathbf{y}} R]$ is supposed to be known if the calculations are carried out according to the FDA. Therefore, the operations needed to get $\boldsymbol{\Delta}_{\mathbf{x}} R$ are the operations to calculate the matrix products in (12) plus those to calculate $[\boldsymbol{\Delta}_{\mathbf{x}}\mathbf{y}]$.

Formula (12) can be simplified for a number of special cases. Some of them are discussed below.

6.2. Special case 1: criterion expressed as a sum of individual contributions

If the criterion R is additive, *i.e.* can be presented by a sum of individual contributions, not necessarily quadratic, from the components of the vectors $\mathbf{y}$,

$$R(\mathbf{y}) = \sum_m f(y_m), \tag{13}$$

then the matrix $[\boldsymbol{\Delta}_{\mathbf{y}} R]$ becomes diagonal. An example is the least-squares fit of the calculated data to the experimental ones. The maximal likelihood criterion can also be represented in the form (13) as is discussed below in §8.

If the criterion is not additive but can be expressed as

$$R(\mathbf{y}) = \sum_m f(y_{m1}; y_{m2}; \ldots; y_{mn}) \tag{14}$$

with n being a relatively small number, $[\boldsymbol{\Delta}_{\mathbf{y}} R]$ becomes sparse. Geometric criteria used in crystallography are an example.

6.3. Special case 2: linear dependence variables y and x

When the variables $\mathbf{y}$ depend linearly on the variables $\mathbf{x}$, $\mathbf{y} = [A]\mathbf{x}$, the second term in the formula is absent and (12) becomes

$$\boldsymbol{\Delta}_{\mathbf{x}} R = [A][\boldsymbol{\Delta}_{\mathbf{y}} R][A]^T. \tag{15}$$

An example of such a linear dependence is the relation between the electron density and its structure factors because Fourier transformation is a linear operation.

6.4. Special case 3: local contribution of the variables x

Quite often variables $\mathbf{x}$ contribute to variables $\mathbf{y}$ locally, *i.e.* each of $x_1, x_2, \ldots, x_N$ contribute only to a small number C_y of variables y_k, $C_y \ll M$. Complementarily, every y_k may depend only on a small number of C_x parameters x_j, $C_x \ll N$. An example of such dependence is the calculation of any field (*e.g.* an electron density) from an atomic model where atoms have a limited radius of contribution and are separated from each other. The matrix $[\mathrm{d}\mathbf{y}/\mathrm{d}\mathbf{x}]$ becomes sparse giving an essential reduction in the number of calculations.

7. Crystallographic least-squares refinement, individual atoms

It is easy to see that the calculation of a Hessian matrix for crystallographic criteria can profit the features of all particular cases discussed above. Similarly to Tronrud (1999), the model structure factors are supposed to be already calculated. In this section, we estimate the number of operations necessary to calculate the matrix of the second derivatives of the least-squares criterion with respect to individual atomic coordinates following the chain rule.

7.1. Step 1: matrix with respect to structure factors

The least-squares criterion R is a sum of individual contributions from M independent structure factors F:

$$R = R(\mathbf{F}) = \sum_j w_j f(F_j), \tag{16}$$

where $f(F_j) = (|F_j| - F_{oj})^2$ and the one-dimensional index j represents a sequential number of a structure factor in the list. The calculation of the matrix $[\boldsymbol{\Delta}_{\mathbf{F}} R]$, which is a diagonal (see §6.2) needs C_1M operations. Each diagonal element is

$$H_{F,j} = [\boldsymbol{\Delta}_F R]_{jj} = \sum_j w_j \partial^2 f(F_j)/\partial F_j^2. \tag{17}$$

It should be remembered that in fact F_j are complex numbers and $H_{F,j}$, formally speaking, is a 2×2 matrix when represented by real variables as happens in computers. This problem for structure factors obeying Hermitian symmetry has been treated by Lunin & Urzhumtsev (1985) where it was also shown that for the derivative calculation a representation of these complex numbers by their real and imaginary parts is more efficient than the usual crystallographic representation by their modulus and phase. We avoid repeating technical details of this analysis.

7.2. Step 2: matrix with respect to density distribution

Since structure factors and electron density are related by the Fourier transform

$$y_m = \sum_j [x_j \exp(i2\pi \mathbf{s}_m \mathbf{r}_j)], \tag{18}$$

which is a linear operation, the second component of the sum (12) is absent as is shown in §6.3. Here $\mathbf{s}_k$ and $\mathbf{r}_j$ are the coordinates of the variables y_m and x_j, respectively, expressed in corresponding coordinate systems. If $\mathbf{y}$ corresponds to a set of structure-factor values and $\mathbf{x}$ corresponds to a set of density values calculated at a grid, then $\mathbf{s}_m$ represents the Miller indices h, k, l, a complex number y_m is the value of this structure factor, and every grid point j has its coordinates $\mathbf{r}_j$ and the density value x_j. For the transformation of electron density into structure factors $\mathbf{F} = [A]\boldsymbol{\rho}$, the matrix $[A]$ in (15) becomes

$$[\mathbf{A}] = [\mathrm{d}\mathbf{F}/\mathrm{d}\boldsymbol{\rho}]$$
$$= \begin{pmatrix} \exp(i2\pi \mathbf{s}_1 \mathbf{r}_1) & \exp(i2\pi \mathbf{s}_2 \mathbf{r}_1) & \dots & \exp(i2\pi \mathbf{s}_M \mathbf{r}_1) \\ \exp(i2\pi \mathbf{s}_1 \mathbf{r}_2) & \exp(i2\pi \mathbf{s}_2 \mathbf{r}_2) & \dots & \exp(i2\pi \mathbf{s}_M \mathbf{r}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \exp(i2\pi \mathbf{s}_1 \mathbf{r}_K) & \exp(i2\pi \mathbf{s}_2 \mathbf{r}_K) & \dots & \exp(i2\pi \mathbf{s}_M \mathbf{r}_K) \end{pmatrix}. \tag{19}$$

With (17), the matrix product (15) becomes (see Appendix A)

$$[\Delta_{\boldsymbol{\rho}} R] = \begin{pmatrix} h(\mathbf{r}_1 + \mathbf{r}_1) & h(\mathbf{r}_2 + \mathbf{r}_1) & \dots & h(\mathbf{r}_K + \mathbf{r}_1) \\ h(\mathbf{r}_1 + \mathbf{r}_2) & h(\mathbf{r}_2 + \mathbf{r}_2) & \dots & h(\mathbf{r}_K + \mathbf{r}_2) \\ \vdots & \vdots & \vdots & \vdots \\ h(\mathbf{r}_1 + \mathbf{r}_K) & h(\mathbf{r}_2 + \mathbf{r}_K) & \dots & h(\mathbf{r}_K + \mathbf{r}_K) \end{pmatrix}, \tag{20}$$

where all elements of the matrix (20) are presented by the same function

$$h(\mathbf{r}) = \sum_m H_{F,m} \exp(i2\pi \mathbf{s}_m, \mathbf{r}) \tag{21}$$

calculated at different points $\mathbf{r}$. In order to calculate this function in a grid compatible with the number of Fourier coefficients M, $K \sim M$, the number of operations is $C_2 M \ln M$ (Cooley & Tukey, 1965; Ten Eyck, 1973).

7.3. Step 3: matrix with respect to atomic parameters

The calculation of the electron-density distribution from an atomic model is a particular case of a local dependence discussed above (§6.4). In this case, C_y is the number of grid points where an atom contributes (it is proportional to the cube of the ratio of the atomic radius to the grid step) and C_x is the number of atoms contributing to every grid point. For usual values of atomic parameters and the grid step, the order of these constants is about 100 and 10, respectively. If the gradient calculation is carried out following FDA and does not need any extra operation, then the number of operations for the transition from $[\Delta_{\boldsymbol{\rho}} R]$ to the matrix $[\Delta_{\mathbf{q}} R]$ with respect to the atomic parameters could be estimated as follows.

The first term in (12) corresponds to the matrix

$$\begin{pmatrix} \sum\limits_{j,k} h(\mathbf{r}_j + \mathbf{r}_k) \frac{\partial \rho_j}{\partial q_1} \frac{\partial \rho_k}{\partial q_1} & \sum\limits_{j,k} h(\mathbf{r}_j + \mathbf{r}_k) \frac{\partial \rho_j}{\partial q_2} \frac{\partial \rho_k}{\partial q_1} & \dots & \sum\limits_{j,k} h(\mathbf{r}_j + \mathbf{r}_k) \frac{\partial \rho_j}{\partial q_N} \frac{\partial \rho_k}{\partial q_1} \\ \sum\limits_{j,k} h(\mathbf{r}_j + \mathbf{r}_k) \frac{\partial \rho_j}{\partial q_1} \frac{\partial \rho_k}{\partial q_2} & \sum\limits_{j,k} h(\mathbf{r}_j + \mathbf{r}_k) \frac{\partial \rho_j}{\partial q_2} \frac{\partial \rho_k}{\partial q_2} & \dots & \sum\limits_{j,k} h(\mathbf{r}_j + \mathbf{r}_k) \frac{\partial \rho_j}{\partial q_N} \frac{\partial \rho_k}{\partial q_2} \\ \vdots & \vdots & \vdots & \vdots \\ \sum\limits_{j,k} h(\mathbf{r}_j + \mathbf{r}_k) \frac{\partial \rho_j}{\partial q_1} \frac{\partial \rho_k}{\partial q_N} & \sum\limits_{j,k} h(\mathbf{r}_j + \mathbf{r}_k) \frac{\partial \rho_j}{\partial q_2} \frac{\partial \rho_k}{\partial q_N} & \dots & \sum\limits_{j,k} h(\mathbf{r}_j + \mathbf{r}_k) \frac{\partial \rho_j}{\partial q_N} \frac{\partial \rho_k}{\partial q_N} \end{pmatrix}. \tag{22}$$

For each of its N^2 elements, only about C_y^2 derivatives are different from 0 from the total number of K^2 giving the estimation of $C_y^2 N^2$ operations.

The second term in (13) corresponds to elements

$$\partial^2 R/\partial q_j \partial q_k = \sum_m [(\partial R/\partial \rho_m)(\partial^2 \rho_m/\partial q_j \partial q_k)]. \tag{23}$$

Similarly to the previous consideration, each of them has no more than C_y members $\partial^2 \rho_m/\partial q_j \partial q_k$ different from 0 giving the estimation by $C_y N^2$ operations. This number can be smaller if all elements (23) can be kept in memory simultaneously (for example, in the cycle over q_j, for every variable ρ_m to which the contribution from q_j is non zero, other L_x variables contributing to q_k are determined and the contribution is added to the matrix elements, the number of operations being proportional to NC_xC_y).

In total, the number of operations for the transition $[\Delta_{\boldsymbol{\rho}} R] \rightarrow [\Delta_{\mathbf{q}} R]$ is of the order of N^2.

7.4. Total time for the exact matrix calculation

Summarizing, if the chain recalculation of the Hessian matrix is used, the total number of operations necessary to calculate the *exact* matrix of the second derivatives of the crystallographic least-squares criterion with respect to the atomic parameters is

$$T_{HC} = C_1 M + C_2 M \ln M + C_3 N^2 \sim C_{12} M \ln M + C_3 N^2, \tag{24}$$

where C_1, C_2, C_3 and C_{12} are some constants that depend neither on the number of structure factors M nor on the number of atoms N. This estimate is the same as (3) for an *approximate* matrix where the second-order terms $\partial^2 R/\partial F^2$ are neglected from the beginning and is better than (9), which is obtained by simple N-times calculation of the product of a matrix by a direction following a coordinate axis.

8. Other crystallographic criteria

One more advantage of the chain scheme for matrix calculation is the same as for the gradient calculation: any transition step 'knows nothing' of how the initial gradient or matrix were obtained and how they will be used further. This allows one to analyse the influence of every transition step and relevant algorithms independently of other steps.

An important feature of the least-squares criterion is that the matrix of its second derivatives with respect to the structure factors as complex numbers is a diagonal one. This means that for any other criterion of the same type (13) as the least-squares one, the time for the matrix calculation will be the same.

8.1. Intensity least-squares criterion

In *SHELX* (Sheldrick & Schneider, 1997), the least-squares technique is applied to structure-factor intensities and not to the magnitudes. This criterion can be treated exactly in the same way as is discussed above.

8.2. Phase criterion

The phase criterion introduced by Lunin & Urzhumtsev (1985) can be used in the same way if the probability distribution coefficients A, B, C and D are considered as constants. In the procedure suggested by Pannu *et al.* (1998), they are iteratively recalculated from the atomic coordinates using the R-free technique suggested by Brünger (1992) and applied for the phase-quality estimation by Lunin & Skovoroda (1995), Pannu & Read (1996), Read (1997) and Pannu *et al.* (1998). However, one can expect that the variation of these coefficients is negligible, the coefficients can be considered as constant, at least for several consecutive minimization steps. As a consequence, the computational scheme developed for the least-squares criterion can be conserved for this criterion.

8.3. Maximum-likelihood criterion

Similar arguments can be applied for the maximum-likelihood criterion. This criterion, when the parameters for the probability are fixed, takes slightly more time for the calculation of its derivatives with respect to the structure factors because of the more complicated function $f(F)$ but other calculations are exactly the same. Additionally, this criterion can be represented in the least-squares form where structure-factor magnitudes calculated from a model are fitted to some *modified* experimental values (see *e.g.* Lunin & Urzhumtsev, 1999). In other words, an explicit quadratic approximation to this criterion can be built. The variation of the magnitude modification is expected to be small (while the modifications themselves can be important), the modification parameters could be considered as constants for many refinement cycles and the situation is reduced to the least-squares case.

8.4. General structure-factor-based criterion

In the general situation when the criterion depends on all structure factors together and cannot be represented by a sum of contributions from many small subsets of structure factors, the Hessian matrix with respect to the structure factors $[\boldsymbol{\Delta}_{\mathbf{F}}R]$ is no longer diagonal and the first component C_1M in the total estimation (24) should be replaced by the C_1M^2. While the last component in (24) does not depend at all on the structure-factor criterion and corresponding estimations can be kept as they are, the second term, corresponding to the transition from structure factors to the electron density, $C_2M \ln M$, also becomes larger. Now the product $[\mathrm{d}\mathbf{F}/\mathrm{d}\boldsymbol{\rho}][\boldsymbol{\Delta}_{\mathbf{F}}R][\mathrm{d}\mathbf{F}/\mathrm{d}\boldsymbol{\rho}]^T$ becomes (see Appendix B)

$$\begin{pmatrix} \sum\limits_m h_m(\mathbf{r}_1)\exp(i2\pi\mathbf{s}_m\mathbf{r}_1) & \sum\limits_m h_m(\mathbf{r}_2)\exp(i2\pi\mathbf{s}_m\mathbf{r}_1) & \ldots & \sum\limits_m h_m(\mathbf{r}_K)\exp(i2\pi\mathbf{s}_m\mathbf{r}_1) \\ \sum\limits_m h_m(\mathbf{r}_1)\exp(i2\pi\mathbf{s}_m\mathbf{r}_2) & \sum\limits_m h_m(\mathbf{r}_2)\exp(i2\pi\mathbf{s}_m\mathbf{r}_2) & \ldots & \sum\limits_m h_m(\mathbf{r}_K)\exp(i2\pi\mathbf{s}_m\mathbf{r}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \sum\limits_m h_m(\mathbf{r}_1)\exp(i2\pi\mathbf{s}_m\mathbf{r}_K) & \sum\limits_m h_m(\mathbf{r}_2)\exp(i2\pi\mathbf{s}_m\mathbf{r}_K) & \ldots & \sum\limits_m h_m(\mathbf{r}_K)\exp(i2\pi\mathbf{s}_m\mathbf{r}_K) \end{pmatrix}, \tag{25}$$

where the M functions

$$h_m(\mathbf{r}) = \sum_j H_{F,mj}\exp[i2\pi(\mathbf{s}_j, \mathbf{r})] \tag{26}$$

are calculated at K different points of the grid as the Fourier transformation by $M(C_5M\ln M) = C_5M^2\ln M$ operations. Elements of every column, $1 \le k \le K$, in (25) are calculated as a Fourier series of the function $g_k(\mathbf{s}_m) = h_m(\mathbf{r}_k)$, obtained at M points $\mathbf{s}_m$, $1 \le m \le M$, of reciprocal space. Being calculated for all K columns, this takes again of the order of $KM\ln M \sim M^2\ln M$ operations. The total number of computer operations becomes

$$T_{HG} = C_6M^2\ln M + C_3N^2. \tag{27}$$

9. Other molecular models

The described procedure can be applied for any type of atomic model: isotropic, anisotropic, multipole models (see Hansen & Coppens, 1978) *etc*. It can also be extended to other types of geometrical objects used for the diffraction simulation instead of independent atoms (see, for example, Kalinin, 1981).

If the independent parameters are not atomic ones (*e.g.* parameters for a rigid-group refinement), the matrix for such non-atomic models can be calculated in the same way. Owing to chain calculations, all previous steps are conserved and one more step is added corresponding to the last recalculation of the matrix $\boldsymbol{\Delta}_{\mathbf{q}}R \rightarrow \boldsymbol{\Delta}_{\boldsymbol{\chi}}R$:

$$\boldsymbol{\Delta}_{\boldsymbol{\chi}}R = [\mathrm{d}\mathbf{q}/\mathrm{d}\boldsymbol{\chi}][\boldsymbol{\Delta}_{\mathbf{q}}R][\mathrm{d}\mathbf{q}/\mathrm{d}\boldsymbol{\chi}]^T + [\boldsymbol{\Delta}_{\boldsymbol{\chi}}\mathbf{q}]\nabla_{\mathbf{q}}R. \tag{28}$$

The larger the blocks of related parameters, the more time is necessary for the computation of (28). The same scheme can be applied when generalized parameters are used not for better molecular description but for better convergence of the minimization process [*e.g.* Tronrud (1992); a similar approach was independently developed and realised in *FROG* by Urzhumtsev *et al.* (1989)].

10. Direct calculation of the inverted Hessian matrix

Formula (12) also gives an idea that for some special cases the inverted matrix of the second derivatives can be obtained directly without calculation of the Hessian matrix itself. Indeed, if the transformation of $\mathbf{x}$ to $\mathbf{y}$ is linear, then it follows from (15) that

$$[\boldsymbol{\Delta}_{\mathbf{x}}R]^{-1} = [A^{-1}]^T[\boldsymbol{\Delta}_{\mathbf{y}}R]^{-1}[A^{-1}]. \tag{29}$$

If $[A]$ corresponds to the Fourier transformation, the inverse operation is the inverse Fourier transform for which the matrix $[A^{-1}]$ can be written immediately. The matrix $[\boldsymbol{\Delta}_{\mathbf{F}}R]^{-1}$ can be easily calculated for many crystallographic criteria, in particular for such important criteria as the least-squares or maximum-likelihood functionals. Therefore, when the independent parameters are density values at the grid points, the inverse Hessian matrix can be easily and directly calculated for these criteria as (see Appendix *C*):

$$[\Delta_\rho R] = \begin{pmatrix} u(\mathbf{r}_1+\mathbf{r}_1) & u(\mathbf{r}_2+\mathbf{r}_1) & \ldots & u(\mathbf{r}_K+\mathbf{r}_1) \\ u(\mathbf{r}_1+\mathbf{r}_2) & u(\mathbf{r}_2+\mathbf{r}_2) & \ldots & u(\mathbf{r}_K+\mathbf{r}_2) \\ \vdots & \vdots & \vdots & \vdots \\ u(\mathbf{r}_1+\mathbf{r}_K) & u(\mathbf{r}_2+\mathbf{r}_K) & \ldots & u(\mathbf{r}_K+\mathbf{r}_K) \end{pmatrix}. \quad (30)$$

Here again, all elements of the matrix (30) are presented by the same function

$$u(\mathbf{r}) = \sum_m U_{F,m} \exp(-i2\pi\mathbf{s}_m, \mathbf{r}) \quad (31)$$

calculated at different points $\mathbf{r}$ ($U_{F,m}$ is the product of some constant with the *m*-diagonal element of this inverted Hessian matrix $[\boldsymbol{\Delta}_{\mathbf{F}}R]^{-1}$ of the crystallographic criterion with respect to structure factors). In order to calculate this function at a grid compatible with the number of Fourier coefficients *M*, estimating $K \sim M$, the number of operations needed is about $C_2M \ln M$ (Cooley & Tukey, 1965; Ten Eyck, 1973). This result shows that the minimization methods of simple iteration, usually applied to density-modification procedures, can be replaced not only by the gradient methods [Sayre (1972) and Sayre & Toupin (1975) for the particular Sayre criterion; Lunin (1985) for the general case] but even by the methods of second order. In this case, the computational expenses are practically the same as those for the simple iteration methods.

11. Discussion

An optimal algorithm for the calculation of the *exact* matrix of second derivatives is suggested for crystallographic criteria and the estimations of the CPU time necessary to calculate it are obtained. In the case of the crystallographic least-squares refinement of an atomic model, the number of computer operations is

$$T_{HC} = C_{12}M \ln M + C_3N^2, \quad (32)$$

where *M* is the number of structure factors, *N* is the number of atomic parameters and C_{12} and C_3 are some constants that do not depend on either *M* or *N*. This estimation is the same as that obtained for an *approximate* matrix calculation neglecting the second-order terms of the least-squares criterion with respect to the structure factors (Tronrud, 1999). Most other known crystallographic diffraction criteria, for example, the maximum-likelihood criterion, need the same order of computer operations.

This algorithm suggests step-by-step recalculation of the matrix with respect to variables of different levels of the molecular models (structure factors, density, atomic parameters *etc.*). The same iterative calculations are basic steps for the fast gradient calculation (Lunin & Urzhumtsev, 1985). It should be noted that such a method of calculation allows the contribution from any other criterion of the same type or of any other type of model to be added, *e.g.* phase criterion, stereochemical criteria, criteria depending on the electron density *etc.* Therefore, the formulae that give the expression for the gradient (or for the Hessian matrix) of a crystallographic criterion directly in terms of atomic parameters can be useful for understanding but may be rather misleading algorithmically.

The suggested algorithms do not explain how, in the general case, to calculate rapidly the *inverted* Hessian matrix that is needed in the minimization procedures of the second order. Such a procedure can be suggested only for some special but important cases as density refinement. It can be noted that in fact these methods do not need the matrix itself but its product by a given vector, for example the gradient. This allows the hope that some new numerical methods can be developed that will be roughly linear on the number of parameters and which will not need N^2 memory elements to keep the matrix, similar to the calculation of the product of the matrix by a given vector following the fast differentiation algorithm.

APPENDIX *A*
A matrix with respect to the density distribution

Let *R* be an additive criterion of structure factors and $H_{F,j}$ stand for the *jj* element of the diagonal matrix $[\boldsymbol{\Delta}_{\mathbf{F}}R]$. With (19),

$$[\Delta_{\mathbf{F}}R][\mathrm{d}\mathbf{F}/\mathrm{d}\boldsymbol{\rho}]^T = \begin{pmatrix} H_{F,1}\exp(i2\pi\mathbf{s}_1\mathbf{r}_1) & H_{F,1}\exp(i2\pi\mathbf{s}_1\mathbf{r}_2) & \ldots & H_{F,1}\exp(i2\pi\mathbf{s}_1\mathbf{r}_K) \\ H_{F,2}\exp(i2\pi\mathbf{s}_2\mathbf{r}_1) & H_{F,2}\exp(i2\pi\mathbf{s}_2\mathbf{r}_2) & \ldots & H_{F,2}\exp(i2\pi\mathbf{s}_2\mathbf{r}_K) \\ \vdots & \vdots & \vdots & \vdots \\ H_{F,M}\exp(i2\pi\mathbf{s}_M\mathbf{r}_1) & H_{F,M}\exp(i2\pi\mathbf{s}_M\mathbf{r}_2) & \ldots & H_{F,M}\exp(i2\pi\mathbf{s}_M\mathbf{r}_K) \end{pmatrix}, \quad (33)$$

where $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_K$ is a list of vectors corresponding to the grid points in real space. The full matrix product (15) is

$$[\Delta_\rho R] = [\mathrm{d}\mathbf{F}/\mathrm{d}\boldsymbol{\rho}][\Delta_{\mathbf{F}}R][\mathrm{d}\mathbf{F}/\mathrm{d}\boldsymbol{\rho}]^T = \begin{pmatrix} \sum\limits_m H_{F,m}\exp[i2\pi\mathbf{s}_m(\mathbf{r}_1+\mathbf{r}_1)] & \sum\limits_m H_{F,m}\exp[i2\pi\mathbf{s}_m(\mathbf{r}_2+\mathbf{r}_1)] & \ldots & \sum\limits_m H_{F,m}\exp[i2\pi\mathbf{s}_m(\mathbf{r}_K+\mathbf{r}_1)] \\ \sum\limits_m H_{F,m}\exp[i2\pi\mathbf{s}_m(\mathbf{r}_1+\mathbf{r}_2)] & \sum\limits_m H_{F,m}\exp[i2\pi\mathbf{s}_m(\mathbf{r}_2+\mathbf{r}_2)] & \ldots & \sum\limits_m H_{F,m}\exp[i2\pi\mathbf{s}_m(\mathbf{r}_K+\mathbf{r}_2)] \\ \vdots & \vdots & \vdots & \vdots \\ \sum\limits_m H_{F,m}\exp[i2\pi\mathbf{s}_m(\mathbf{r}_1+\mathbf{r}_K)] & \sum\limits_m H_{F,m}\exp[i2\pi\mathbf{s}_m(\mathbf{r}_2+\mathbf{r}_K)] & \ldots & \sum\limits_m H_{F,m}\exp[i2\pi\mathbf{s}_m(\mathbf{r}_K+\mathbf{r}_K)] \end{pmatrix}. \quad (34)$$

All elements of matrix (34) can be calculated as a Fourier series at points $\mathbf{r}$:

$$h(\mathbf{r}) = \sum_m H_{F,m} \exp(i2\pi\mathbf{s}_m, \mathbf{r}), \tag{35}$$

which finally gives (20).

APPENDIX *B*
General structure-factor-based criterion

In the general case when the criterion depends on all structure factors together and cannot be represented as a sum of contributions from many small subsets of structure factors, the matrix of the structure factors is no longer special. The product $[\boldsymbol{\Delta}_\mathbf{F}R][\mathrm{d}\mathbf{F}/\mathrm{d}\boldsymbol{\rho}]^T$ becomes equal to

$$\begin{pmatrix} \sum\limits_j H_{F,1j}\exp(i2\pi\mathbf{s}_j\mathbf{r}_1) & \sum\limits_j H_{F,1j}\exp(i2\pi\mathbf{s}_j\mathbf{r}_2) & \ldots & \sum\limits_j H_{F,1j}\exp(i2\pi\mathbf{s}_j\mathbf{r}_K) \\ \sum\limits_j H_{F,2j}\exp(i2\pi\mathbf{s}_j\mathbf{r}_1) & \sum\limits_j H_{F,2j}\exp(i2\pi\mathbf{s}_j\mathbf{r}_2) & \ldots & \sum\limits_j H_{F,2j}\exp(i2\pi\mathbf{s}_j\mathbf{r}_K) \\ \vdots & \vdots & \vdots & \vdots \\ \sum\limits_j H_{F,Mj}\exp(i2\pi\mathbf{s}_j\mathbf{r}_1) & \sum\limits_j H_{F,Mj}\exp(i2\pi\mathbf{s}_j\mathbf{r}_2) & \ldots & \sum\limits_j H_{F,Mj}\exp(i2\pi\mathbf{s}_j\mathbf{r}_K) \end{pmatrix} \tag{36}$$

or

$$\begin{pmatrix} h_1(\mathbf{r}_1) & h_1(\mathbf{r}_2) & \ldots & h_1(\mathbf{r}_K) \\ h_2(\mathbf{r}_1) & h_2(\mathbf{r}_2) & \ldots & h_2(\mathbf{r}_K) \\ \vdots & \vdots & \vdots & \vdots \\ h_M(\mathbf{r}_1) & h_M(\mathbf{r}_2) & \ldots & h_M(\mathbf{r}_K) \end{pmatrix}, \tag{37}$$

where elements of every line m represent the function $h_m(\mathbf{r})$ calculated at K different points of the grid. These functions are calculated as the Fourier transformation

$$h_m(\mathbf{r}) = \sum_j H_{F,mj} \exp[i2\pi(\mathbf{s}_j, \mathbf{r})]. \tag{38}$$

The multiplication $[\mathrm{d}\mathbf{F}/\mathrm{d}\boldsymbol{\rho}]$ by (37) gives $[\mathrm{d}\mathbf{F}/\mathrm{d}\boldsymbol{\rho}][\boldsymbol{\Delta}_\mathbf{F}R][\mathrm{d}\mathbf{F}/\mathrm{d}\boldsymbol{\rho}]^T$ in the form (25).

APPENDIX *C*
Direct calculation of the inverted Hessian matrix

If the matrix $[\mathbf{A}]$ corresponds to the Fourier transformation (18), where $\mathbf{y}$ are structure factors and $\mathbf{x}$ are density values, the inverse operation is the inverse Fourier transform for which the matrix is

$$\begin{aligned}[\mathbf{A}^{-1}] &= [\mathrm{d}\mathbf{y}/\mathrm{d}\mathbf{x}]^{-1} \\ &= d_V \begin{pmatrix} \exp(-i2\pi\mathbf{s}_1\mathbf{r}_1) & \exp(-i2\pi\mathbf{s}_1\mathbf{r}_2) & \ldots & \exp(-i2\pi\mathbf{s}_1\mathbf{r}_K) \\ \exp(-i2\pi\mathbf{s}_2\mathbf{r}_1) & \exp(-i2\pi\mathbf{s}_2\mathbf{r}_2) & \ldots & \exp(-i2\pi\mathbf{s}_2\mathbf{r}_K) \\ \vdots & \vdots & \vdots & \vdots \\ \exp(-i2\pi\mathbf{s}_M\mathbf{r}_1) & \exp(-i2\pi\mathbf{s}_M\mathbf{r}_2) & \ldots & \exp(-i2\pi\mathbf{s}_M\mathbf{r}_K) \end{pmatrix},\end{aligned} \tag{39}$$

where the coefficient $d_V = V/(N_xN_yN_z)$, V being the volume of the unit cell and N_x, N_y, N_z the grid numbers. If $U_{F,j}$ stands for the jj diagonal element of the matrix $[\boldsymbol{\Delta}_\mathbf{y}R]^{-1}$, then

$$\begin{aligned}&[\Delta_\mathbf{F}R]^{-1}[\mathrm{d}\mathbf{F}/\mathrm{d}\boldsymbol{\rho}]^{-1} \\ &= \frac{1}{d_V}\begin{pmatrix} U_{F,1}\exp(-i2\pi\mathbf{s}_1\mathbf{r}_1) & U_{F,1}\exp(-i2\pi\mathbf{s}_1\mathbf{r}_2) & \ldots & U_{F,1}\exp(-i2\pi\mathbf{s}_1\mathbf{r}_K) \\ U_{F,2}\exp(-i2\pi\mathbf{s}_2\mathbf{r}_1) & U_{F,2}\exp(-i2\pi\mathbf{s}_2\mathbf{r}_2) & \ldots & U_{F,2}\exp(-i2\pi\mathbf{s}_2\mathbf{r}_K) \\ \vdots & \vdots & \vdots & \vdots \\ U_{F,M}\exp(-i2\pi\mathbf{s}_M\mathbf{r}_1) & U_{F,M}\exp(-i2\pi\mathbf{s}_M\mathbf{r}_2) & \ldots & U_{F,M}\exp(-i2\pi\mathbf{s}_M\mathbf{r}_K) \end{pmatrix}\end{aligned} \tag{40}$$

with

$$[U] = d_V^2[\Delta_\mathbf{F}R]^{-1} \tag{41}$$

and finally

$$\begin{aligned}[\Delta_{\boldsymbol{\rho}}R]^{-1} &= \big[[\mathrm{d}\mathbf{F}/\mathrm{d}\boldsymbol{\rho}]^{-1}\big]^T[\Delta_\mathbf{F}R]^{-1}[\mathrm{d}\mathbf{F}/\mathrm{d}\boldsymbol{\rho}]^{-1} \\ &= \begin{pmatrix} \sum\limits_m U_{F,m}\exp[-i2\pi\mathbf{s}_m(\mathbf{r}_1+\mathbf{r}_1)] & \sum\limits_m U_{F,m}\exp[-i2\pi\mathbf{s}_m(\mathbf{r}_2+\mathbf{r}_1)] & \ldots & \sum\limits_m U_{F,m}\exp[-i2\pi\mathbf{s}_m(\mathbf{r}_K+\mathbf{r}_1)] \\ \sum\limits_m U_{F,m}\exp[-i2\pi\mathbf{s}_m(\mathbf{r}_1+\mathbf{r}_2)] & \sum\limits_m U_{F,m}\exp[-i2\pi\mathbf{s}_m(\mathbf{r}_2+\mathbf{r}_2)] & \ldots & \sum\limits_m U_{F,m}\exp[-i2\pi\mathbf{s}_m(\mathbf{r}_K+\mathbf{r}_2)] \\ \vdots & \vdots & \vdots & \vdots \\ \sum\limits_m U_{F,m}\exp[-i2\pi\mathbf{s}_m(\mathbf{r}_1+\mathbf{r}_K)] & \sum\limits_m U_{F,m}\exp[-i2\pi\mathbf{s}_m(\mathbf{r}_2+\mathbf{r}_K)] & \ldots & \sum\limits_m U_{F,m}\exp[-i2\pi\mathbf{s}_m(\mathbf{r}_K+\mathbf{r}_K)] \end{pmatrix}.\end{aligned} \tag{42}$$

Again, all elements of the matrix (42) are presented by the same function

$$u(\mathbf{r}) = \sum_m U_{F,m} \exp(-i2\pi\mathbf{s}_m, \mathbf{r}) \tag{43}$$

calculated at different points $\mathbf{r}$ thus giving (30).

This work was supported by RFBR grant 00-04-48175 (VYL). The authors thank A. G. Kushnirenko and K. V. Kim who attracted their attention to the fast differentiation algorithm and C. Lecomte and the Centre Charles Hermite (Nancy) for their support of the work.

References

Agarwal, R. C. (1978). *Acta Cryst.* A**34**, 791–809.

Agarwal, R. C. (1981). *Refinement of Protein Structures*. Proceedings of the Daresbury Study Weekend, 15–16 November 1980; compiled by P. A. Machin, J. W. Campbell & M. Elder, pp. 24–28. SERC Daresbury Laboratory, Warrington, England.

Baur, W. & Strassen, V. (1983). *Theor. Comput. Sci.* **22**, 317–330.

Brünger, A. T. (1992). *Nature (London)*, **355**, 472–474.

Cooley, J. W. & Tukey, J. W. (1965). *Math. Comput.* **19**, 297–301.

Cowtan, K. & Ten Eyck, L. F. (1998). *ECM-18 Abstracts*, XVIIIth European Crystallographic Meeting, 16–20 August 1998, Prague, Czeck Republic, E5-P7. *Bulletin of the Czeck and Slovak Crystallographic Association*, **5**A, 136.

Dodson, E. (1981). *Macromolecular Refinement*. Proceedings of the CCP4 Study Weekend, edited by E. Dodson, M. Moore, A. Ralph & S. Bailey, pp. 85–92. Daresbury Laboratory, Warrington, England.

Guillot, B., Viry, L., Guillot, R., Lecomte, C. & Jelsch, C. (2001). *J. Appl. Cryst.* **34**, 214–222.

Hansen, N. K. & Coppens, P. (1978). *Acta Cryst.* A**34**, 909–921.

Hendrickson, W. A. & Konnert, J. H. (1980). *Biomolecular Structure, Function, Conformation and Evolution*, edited by R. S. Srinivasan, Vol. 1, pp. 43–57. Oxford: Pergamon.

Hestenes, M. R. & Stiefel, E. (1952). *J. Res. Natl Bur. Stand.* **49**, 409–436.

Kalinin, D. I. (1981). *Kristallographiya*, **25**, 535–544.

Kim, K. M., Nesterov, Yu. E. & Cherkassky, B. V. (1984). *Dokl. Acad. Nauk SSSR*, **275**, 1306–1309.

Lanczos, C. (1952). *J. Res. Natl Bur. Stand.* **49**, 33–53.

Lunin, V. Yu. (1985). *Acta Cryst.* A**41**, 551–556.

Lunin, V. Yu. & Skovoroda, T. P. (1995). *Acta Cryst.* A**51**, 880–887.

Lunin, V. Yu. & Urzhumtsev, A. G. (1985). *Acta Cryst.* A**41**, 327–333.

Lunin, V. Yu. & Urzhumtsev, A. G. (1999). *CCP4 Newslet. Protein Crystallogr.* **37**, 14–28.

Murshudov, G. N., Vagin, A. A. & Dodson, E. J. (1997). *Acta Cryst.* D**53**, 240–255.

Pannu, N. S., Murshudov, G. N., Dodson, E. J. & Read, R. J. (1998). *Acta Cryst.* D**54**, 1285–1294.

Pannu, N. S. & Read, R. J. (1996). *Acta Cryst.* A**52**, 659–668.

Read, R. J. (1997). *Methods Enzymol.* **277**B, 110–128.

Sayre, D. (1951). *Acta Cryst.* **4**, 362–367.

Sayre, D. (1972). *Acta Cryst.* A**28**, 210–212.

Sayre, D. & Toupin, R. A. (1975). *Acta Cryst.* A**31**, S20.

Sheldrick, G. M. & Schneider, T. R. (1997). *Methods Enzymol.* **277**B, 319–343.

Sussman, J. L., Holbrook, S. R., Church, G. M. & Kim, S.-H. (1977). *Acta Cryst.* A**33**, 800–804.

Templeton, D. (1999). *Acta Cryst.* A**55**, 695–699.

Ten Eyck, L. F. (1973). *Acta Cryst.* A**29**, 183–191.

Ten Eyck, L. F. (1977). *Acta Cryst.* A**33**, 486–492.

Ten Eyck, L. F. (1999). *Acta Cryst.* A**55** Supplement, Abstract M08.BB.003.

Tronrud, D. L. (1992). *Acta Cryst.* A**48**, 912–916.

Tronrud, D. L. (1999). *Acta Cryst.* A**55**, 700–703.

Urzhumtsev, A. G., Lunin, V. Yu. & Vernoslova, E. A. (1989). *J. Appl. Cryst.* **22**, 500–506.